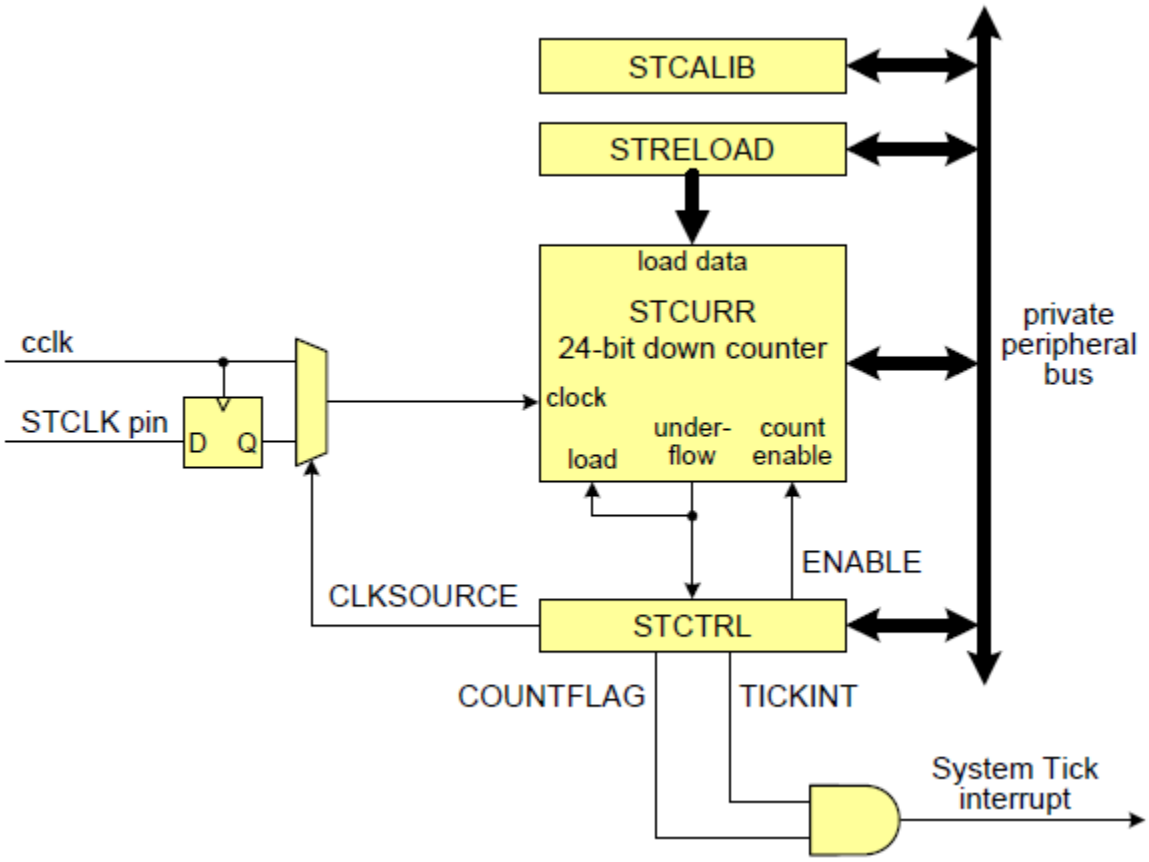
Lesson 14
System Tick Timer (Systick Timer)

1. Overview

In this lesson, the system tick timer (*Systick Timer*) is introduced. *Systick Timer* is a small timer that is integrated as part of the NVIC of the Cortex-M3 processor. This timer is generally used to support an operating system or other system management software. This timer can also be used to create time delays and generate periodic interrupts. Since our microcontroller does not have an operating system, we will mainly use this timer for delays and periodic interrupts. Using a timer for delays (or periodic polling an event) is preferable to using a loop (as in lab 1) because the processor is free to process other tasks while the timer works in the background. There are other timers that are included in our microcontroller, and those timers are left for you to explore. We will focus on the *Systick Timer* in this lesson.

2. Systick Timer

The main component of the *Systick Timer* is a 24-bit down counter that can be operated with the processor clock or an input clock signal from a pin (*STCLK* pin). The general block diagram of a *Systick Timer* is shown in figure below. The *Systick Timer* is intended to generate fixed *10-ms* interrupts to support a system management software (e.g. an OS). However, the timer can also be used as a general purpose timer.



From *Fig. 118 in the LPC17xx User manual*, NXP Semiconductors, 2010.

The can be controlled using four registers:

Name	Description	Access	Reset value[1]	Address
STCTRL	System Timer Control and status register	R/W	0x4	0xE000 E010
STRELOAD	System Timer Reload value register	R/W	0	0xE000 E014
STCURR	System Timer Current value register	R/W	0	0xE000 E018
STCALIB	System Timer Calibration value register	R/W	0x000F 423F	0xE000 E01C

[1] Reset Value reflects the data stored in used bits only. It does not include content of reserved bits.

From *Table 438 in the LPC17xx User manual*, NXP Semiconductors, 2010.

Systick Control and Status Register (STCTRL @ 0xE000E010)

The STCTRL register contains control information for the System Tick Timer, and provides a status flag as shown in table below.

Bit	Symbol	Description	Reset value
0	ENABLE	System Tick counter enable. When 1, the counter is enabled. When 0, the counter is disabled.	0
1	TICKINT	System Tick interrupt enable. When 1, the System Tick interrupt is enabled. When 0, the System Tick interrupt is disabled. When enabled, the interrupt is generated when the System Tick counter counts down to 0.	0
2	CLKSOURCE	System Tick clock source selection. When 1, the CPU clock is selected. When 0, the external clock pin (STCLK) is selected.	1
15:3	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
16	COUNTFLAG	System Tick counter flag. This flag is set when the System Tick counter counts down to 0, and is cleared by reading this register.	0
31:17	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

From *Table 439 in the LPC17xx User manual*, NXP Semiconductors, 2010.

Systick Reload Value Register (STRELOAD @ 0xE000E014)

The value in this register is loaded into the counter whenever the count reaches zero. This value should be initialized before enabling the interrupt.

Bit	Symbol	Description	Reset value
23:0	RELOAD	This is the value that is loaded into the System Tick counter when it counts down to 0.	0
31:24	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

From *Table 440 in the LPC17xx User manual*, NXP Semiconductors, 2010.

Question: What is the maximum value for this register?

Question: What is the value in the register if the recurrent interrupt occurs every 1ms given the 10MHz clock source?

Systick Current Value Register (STCURR@ 0xE000E018)

Reading STCURR register returns the current count from the System Tick counter. Writing to this register clears the current count.

Bit	Symbol	Description	Reset value
23:0	CURRENT	Reading this register returns the current value of the System Tick counter. Writing any value clears the System Tick counter and the COUNTFLAG bit in STCTRL.	0
31:24	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

From *Table 441 in the LPC17xx User manual*, NXP Semiconductors, 2010.

Systick Calibration Value Register (STCALIB@ 0xE000E01C)

The STCALIB register contains a pre-defined value for generating an interrupt every 10 milliseconds if the System Tick Timer is clocked at a frequency of 100 MHz. **The use of this register is not required**.

Bit	Symbol	Value	Description	Reset value
23:0	TENMS		Reload value to get a 10 millisecond System Tick underflow rate when running at 100 MHz. This value initialized at reset with a factory supplied value selected for the LPC17xx. The provided values of TENMS, SKEW, and NOREF are applicable only when using a CPU clock or external STCLK source of 100 MHz.	0x0F 423F
29:24	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
30	SKEW		Indicates whether the TENMS value will generate a precise 10 millisecond time, or an approximation. This bit is initialized at reset with a factory supplied value selected for the LPC17xx. See the description of TENMS above. When 0, the value of TENMS is considered to be precise. When 1, the value of TENMS is not considered to be precise.	0
31	NOREF		Indicates whether an external reference clock is available. This bit is initialized at reset with a factory supplied value selected for the LPC17xx. See the description of TENMS above. When 0, a separate reference clock is available. When 1, a separate reference clock is not available.	0

From *Table 442 in the LPC17xx User manual*, NXP Semiconductors, 2010.

3. Example

Write a C code to configure the *Systick Timer* interrupt that should occur every 1 ms. Assume that the processor clock frequency is 100 MHz.

Tasks:

- *Systick Timer* Setup

- ISR

Question: what is the name of the ISR for *Systick Timer*? How would you declare this ISR in C?

```
/*-------------------------------------------------------------------------
 Lesson 13: using c and Systick timer interrupt
 *-------------------------------------------------------------------------*/
#include "LPC17xx.h"       // Device header

                           // Systick handler declaration

// Define pointers to registers
volatile unsigned int* STCTRL_ptr   = (unsigned int *) 0xE000E010;
volatile unsigned int* STRELOAD_ptr = (unsigned int *) 0xE000E014;
volatile unsigned int* STCURR_ptr   = (unsigned int *) 0xE000E018;

/*-------------------------------------------------------------------------
  Main: Initialize
 *-------------------------------------------------------------------------*/
int main (void) {

     /*Initialize Systick Timer */




     while (1){}               // main loop
}
/*-------------------------------------------------------------------------
 Systick Timer ISR
 *-------------------------------------------------------------------------*/
```

Exercise: Write a C program to for the example above. Use the simulator to verify 1ms interrupts.

4. References

[1]. Joseph Yiu, *The Definitive Guide to ARM Cortex-M3 and Cortex-M4 Processors*, Elsevier, 3rd ed, 2014.
[2]. Jonathan Valvano, *Introduction to ARM Cortex-M Microcontroller*, 4nd ed, 2013.
[3]. *ARMv7-M Architecture Reference Manual*, ARM Limited, 2010.
[4]. *LPC17xx User manual*, NXP Semiconductors, 2010.
[5]. *Cortex-M3 Technical Reference Manual*, ARM Limited, 2010.